Dictionary learning from phaseless measurements

Andreas Tillmann, Yonina Eldar, Julien Mairal

DICTIONARY LEARNING FROM PHASELESS MEASUREMENTS

Andreas M. Tillmann[⋆] *Yonina C. Eldar*[†] *Julien Mairal*[◦]

[⋆] TU Darmstadt, Research Group Optimization, Dolivostr. 15, 64293 Darmstadt, Germany
[†] Department of Electrical Engineering, Technion, Haifa 32000, Israel
[◦] Inria, Lear Team, 655 Avenue de l'Europe, 38330 Montbonnot, France

ABSTRACT

We propose a new algorithm to learn a dictionary along with sparse representations from signal measurements without phase. Specifically, we consider the task of reconstructing a two-dimensional image from squared-magnitude measurements of a complex-valued linear transformation of the original image. Several recent phase retrieval algorithms exploit underlying sparsity of the unknown signal in order to improve recovery performance. In this work, we consider sparse phase retrieval when the sparsifying dictionary is not known in advance, and we learn a dictionary such that each patch of the reconstructed image can be sparsely represented. Our numerical experiments demonstrate that our proposed scheme can obtain significantly better reconstructions for noisy phase retrieval problems than methods that cannot exploit such "hidden" sparsity.

Index Terms— Phase Retrieval, Dictionary Learning

1. INTRODUCTION

Phase retrieval has been an active research topic for decades [1, 2]. In mathematical terms, it can be formulated as

$$\text{find } \boldsymbol{x} \in \mathcal{X} \subseteq \mathbb{C}^N \quad \text{s.t.} \quad |f_i(\boldsymbol{x})|^2 = y_i \quad \forall\, i = 1, \dots, M, \tag{1}$$

where the functions $f_i : \mathbb{C}^N \to \mathbb{C}$ are linear operators and the scalars $y_i = |f_i(\hat{\boldsymbol{x}})|^2$ are nonlinear measurements of the original signal $\hat{\boldsymbol{x}}$ in $\mathcal{X}$, obtained by removing the phase information. A traditional approach is to consider cases in which the solution of (1) is unique, up to global phase, and to devise algorithms to recover $\hat{\boldsymbol{x}}$. Usually, recovering $\hat{\boldsymbol{x}}$ involves oversampling, i.e., taking $M > N$ measurements. The most popular techniques for phase retrieval are based on alternating projections [3, 4, 5]. These methods generally require precise prior information on the signal (such as knowledge of the support set) and often converge to erroneous results. More recent approaches include semidefinite programming [6, 7, 8, 9, 10] and gradient-based methods such as Wirtinger Flow [11, 12].

In recent years, new techniques for (1) were developed when $\hat{\boldsymbol{x}}$ is *sparse*, i.e., is composed of only few atoms from a known dictionary [7, 13, 12, 14], leading to algorithms with improved recovery performance. The main idea is akin to compressed sensing, where one works with fewer *linear* measurements than signal components [15, 16]. An important observation that boosted the applicability of sparse recovery techniques is that many classes of signals admit a sparse approximation in some basis or overcomplete dictionary [17, 18, 19]. While sometimes such dictionaries are known explicitly, better results can often be achieved by adapting the dictionary to the data [18]. Numerous algorithms have been developed for this task (e.g., [20, 21]) when the signal measurements are *linear*.

In this work, we propose a dictionary learning scheme for simultaneously solving the signal reconstruction and sparse representation problems given *nonlinear, phaseless* and possibly *noisy* measurements. We demonstrate its ability to achieve significantly improved reconstructions (especially when the oversampling ratio is low and the noise level high) compared to Wirtinger Flow (WF) [11], which cannot exploit sparsity if the dictionary is unknown. Our algorithm, referred to as phase-retrieval dictionary-learning (PRDL), is based on alternating minimization where we iterate between best fitting the data and sparsely representing the recovered image. PRDL combines projected gradient descent like in Wirtinger Flow to update the image, iterative shrinkage to sparsify the image [22] and block-coordinate descent for the dictionary update [21].

2. PHASE-RETRIEVAL DICTIONARY-LEARNING

Our focus is on the application to image reconstruction. Therefore, we work in the 2D setting directly; however, all expressions and algorithms could easily be formulated for 1D signals, as in (1).

We wish to recover an image $\hat{\boldsymbol{X}}$ in $[0,1]^{N_1 \times N_2}$ from noise-corrupted phaseless nonlinear measurements

$$\boldsymbol{Y} \coloneqq |\mathcal{F}(\hat{\boldsymbol{X}})|^2 + \boldsymbol{N}, \tag{2}$$

where $\mathcal{F} : \mathbb{C}^{N_1 \times N_2} \to \mathbb{C}^{M_1 \times M_2}$ is a linear operator, $\boldsymbol{N}$ denotes noise, and the complex modulus and squares are taken componentwise. Signal sparsity is known to aid in phase retrieval, but a sparsifying transform is not always known a priori. This motivates learning a dictionary $\boldsymbol{D}$ in $\mathbb{R}^{s \times n}$ such that each $s_1 \times s_2$ patch $\hat{\boldsymbol{x}}^i$ of $\hat{\boldsymbol{X}}$, represented as a vector of size $s = s_1 s_2$, can be approximated by $\hat{\boldsymbol{x}}^i \approx \boldsymbol{D}\boldsymbol{a}^i$ with a sparse vector $\boldsymbol{a}^i$ in $\mathbb{R}^n$. Here, n is chosen a priori and the number of patches depends on whether the patches are overlapping or not. In general, $\boldsymbol{D}$ is chosen such that $n \geq s$.

Before presenting our approach for tackling (2), we define the following notation. We consider the linear operator $\mathcal{E} : \mathbb{C}^{N_1 \times N_2} \to \mathbb{C}^{s \times p}$ that extracts the p patches $\boldsymbol{x}^i$ (which may overlap or not) from an image $\boldsymbol{X}$ and forms the matrix $\mathcal{E}(\boldsymbol{X}) = (\boldsymbol{x}^1, \dots, \boldsymbol{x}^p)$. Similarly, we define the linear operator $\mathcal{R} : \mathbb{C}^{s \times p} \to \mathbb{C}^{N_1 \times N_2}$ that reverses this process, i.e., builds an image from a matrix containing vectorized patches as its columns. Thus, in particular, we have $\mathcal{R}(\mathcal{E}(\boldsymbol{X})) = \boldsymbol{X}$. Further, let $\boldsymbol{A} \coloneqq (\boldsymbol{a}^1, \dots, \boldsymbol{a}^p)$ in $\mathbb{R}^{n \times p}$ be the matrix containing the patch representation coefficient vectors as columns. Then, our desired sparse-approximation relation $\boldsymbol{x}^i \approx \boldsymbol{D}\boldsymbol{a}^i$ can be expressed as $\mathcal{E}(\boldsymbol{X}) \approx \boldsymbol{D}\boldsymbol{A}$.

With this notation in hand, we now introduce our method, called phase-retrieval dictionary-learning (PRDL), which aims at solving

$$\min_{\boldsymbol{X},\boldsymbol{D},\boldsymbol{A}} \tfrac{1}{4}\left\|\boldsymbol{Y} - |\mathcal{F}(\boldsymbol{X})|^2\right\|_{\mathrm{F}}^2 + \tfrac{\mu}{2}\left\|\mathcal{E}(\boldsymbol{X}) - \boldsymbol{D}\boldsymbol{A}\right\|_{\mathrm{F}}^2 + \lambda \sum_{i=1}^{p} \left\|\boldsymbol{a}^i\right\|_1$$
$$\text{s.t.} \quad \boldsymbol{X} \in [0,1]^{N_1 \times N_2}, \quad \boldsymbol{D} \in \mathcal{D}. \tag{3}$$

Algorithm 1 Phase-Retrieval Dictionary-Learning Algorithm (PRDL)

Input: initial image estimate $\boldsymbol{X}_{(0)} \in [0,1]^{N_1 \times N_2}$, initial dictionary $\boldsymbol{D}_{(0)} \in \mathcal{D} \subset \mathbb{R}^{s \times n}$, parameters $\mu, \lambda > 0$, iteration limits K_1, K_2;
Output: Learned dictionary $\boldsymbol{D} = \boldsymbol{D}_{(K)}$, patch representations $\boldsymbol{A} = \boldsymbol{A}_{(K)}$, image reconstruction $\boldsymbol{X} = \boldsymbol{X}_{(K)}$;

for $\ell = 0, 1, 2, \ldots, K_1 + K_2$ **do**
  choose step size γ_ℓ^A as explained in Section 3.1 and update $\boldsymbol{A}_{(\ell+1)} \leftarrow \mathcal{S}_{\lambda\gamma_\ell^A/\mu}\Big(\boldsymbol{A}_\ell - \gamma_\ell^A \boldsymbol{D}_{(\ell)}^\top\big(\boldsymbol{D}_{(\ell)}\boldsymbol{A}_{(\ell)} - \mathcal{E}(\boldsymbol{X}_{(\ell)})\big)\Big)$;
  choose step size γ_ℓ^X and update $\boldsymbol{X}_{(\ell+1)} \leftarrow \mathcal{P}_{\mathcal{X}}\Big(\boldsymbol{X}_{(\ell)} - \gamma_\ell^X\Big(\Re\Big(\mathcal{F}^*\big(\mathcal{F}(\boldsymbol{X}) \odot (|\mathcal{F}(\boldsymbol{X})|^2 - \boldsymbol{Y})\big)\Big) + \mu\boldsymbol{R} \odot \mathcal{R}\big(\mathcal{E}(\boldsymbol{X}) - \boldsymbol{D}\boldsymbol{A}\big)\Big)\Big)$;
  if $\ell < K_1$ **then**
    keep $\boldsymbol{D}_{(\ell+1)} \leftarrow \boldsymbol{D}_{(\ell)}$;
  else
    set $\boldsymbol{B} \leftarrow \mathcal{E}(\boldsymbol{X}_{(\ell)})\boldsymbol{A}_{(\ell)}^\top$ and $\boldsymbol{C} \leftarrow \boldsymbol{A}_{(\ell)}\boldsymbol{A}_{(\ell)}^\top$;
    for $j = 1, \ldots, n$ **do**
      if $C_{jj} > 0$ **then**
        update j-th column: $(\boldsymbol{D}_{(\ell+1)})_{\cdot j} \leftarrow \frac{1}{C_{jj}}\big(\boldsymbol{B}_{\cdot j} - \boldsymbol{D}_{(\ell)}\boldsymbol{C}_{\cdot j}\big) + (\boldsymbol{D}_{(\ell)})_{\cdot j}$;
      else
        reset j-th column: e.g., $(\boldsymbol{D}_{(\ell+1)})_{\cdot j} \leftarrow$ random $\mathcal{N}(0,1)$ vector (in $\mathbb{R}^s$);
      normalize $(\boldsymbol{D}_{(\ell+1)})_{\cdot j} \leftarrow \frac{1}{\|(\boldsymbol{D}_{(\ell+1)})_{\cdot j}\|_2}(\boldsymbol{D}_{(\ell+1)})_{\cdot j}$

Here, $\|\boldsymbol{X}\|_{\mathrm{F}}$ denotes the Frobenius matrix-norm, which generalizes the Euclidean norm to matrices. The parameters $\mu, \lambda > 0$ in the objective (3) provide a way to control the trade-off between the data fidelity term from the phase retrieval problem and the approximation sparsity of the image patches. To that effect, we use the ℓ_1-norm, which is well-known to have a sparsity-inducing effect [23]. In order to avoid scaling disambiguities, we also restrict $\boldsymbol{D}$ to be in the subset $\mathcal{D}$ of matrices with unit-ℓ_2-norm columns, and assume $n < p$ (otherwise, each patch is trivially representable by a 1-sparse vector $\boldsymbol{a}^i$ by including $\boldsymbol{x}^i/\|\boldsymbol{x}^i\|_2$ as a column of $\boldsymbol{D}$).

3. ALGORITHMIC FRAMEWORK

Problem (3) is non-convex and difficult to solve, similarly to classical dictionary learning [24, 17, 20, 19]. Therefore, we adopt an algorithm that provides monotonic decrease of the objective. (In fact, convergence to a stationary point of (3) can be obtained under mild assumptions, using results from [25]; we cannot give details here due to space restrictions.)

The algorithmic framework we employ is that of *alternating minimization*: For each variable $\boldsymbol{D}$, $\boldsymbol{A}$ and $\boldsymbol{X}$ in turn, we take one step towards solving (3) w.r.t. this variable alone, keeping the other two fixed. We summarize our method in Algorithm 1, where the superscript * denotes the adjoint operator (for a matrix $\boldsymbol{Z}$, $\boldsymbol{Z}^*$ is thus the conjugate transpose), $\Re(\cdot)$ extracts the real part of a complex-valued argument, and $\odot$ denotes the Hadamard (element-wise) product of two matrices. The algorithm also involves the classical soft-thresholding operator $\mathcal{S}_\tau(\boldsymbol{Z}) \coloneqq \max\{0, |\boldsymbol{Z}| - \tau\} \odot \operatorname{sign}(\boldsymbol{Z})$ and the Euclidean projection $\mathcal{P}_{\mathcal{X}}(\boldsymbol{Z}) \coloneqq \max\{0, \min\{1, \boldsymbol{Z}\}\}$ onto $\mathcal{X} \coloneqq [0,1]^{N_1 \times N_2}$; here, all operations are meant component-wise.

To avoid training the dictionary on potentially useless early estimates, the algorithm performs two phases—while the iteration counter ℓ is smaller than K_1, the dictionary is not updated. We next explain the algorithmic steps in more detail.

3.1. Updating the Patch Representation Vectors

Updating $\boldsymbol{A}$ (i.e., (3) with $\boldsymbol{D}$ and $\boldsymbol{X}$ fixed at their current values) calls for decreasing the objective

$$\sum_{i=1}^{p} \Big(\tfrac{1}{2}\big\|\boldsymbol{D}_{(\ell)}\boldsymbol{a}^i - \boldsymbol{x}_{(\ell)}^i\big\|_2^2 + \tfrac{\lambda}{\mu}\big\|\boldsymbol{a}^i\big\|_1\Big). \tag{4}$$

Since the objective here is separable, we can update all vectors $\boldsymbol{a}^i$ in parallel independently. To do so, we choose to perform one step of the algorithm ISTA (see, e.g., [22]), which has a monotonic decrease property. ISTA is a gradient-based method which performs the following update for each $i = 1, \ldots, p$:

$$\boldsymbol{a}_{(\ell+1)}^i = \mathcal{S}_{\lambda\gamma_\ell^A/\mu}\Big(\boldsymbol{a}_{(\ell)}^i - \gamma_\ell^A \boldsymbol{D}_{(\ell)}^\top\big(\boldsymbol{D}_{(\ell)}\boldsymbol{a}_{(\ell)}^i - \boldsymbol{x}_\ell^i\big)\Big).$$

This update involves a gradient descent step followed by soft-thresholding. The step size parameter γ_ℓ^A can be chosen as $1/L_A$, where L_A is an upper bound on the Lipschitz constant of the gradient; here, $L_A = \|\boldsymbol{D}_{(\ell)}\|_2^2$ would do, but a better strategy is to use a backtracking scheme to automatically update L_A [22].

Constructing $\boldsymbol{A}_{(\ell+1)}$ from the $\boldsymbol{a}_{(\ell+1)}^i$ as specified above is equivalent to Step 2 of Algorithm 1.

3.2. Updating the Image Estimate

With $\boldsymbol{D} = \boldsymbol{D}_{(\ell)}$ and $\boldsymbol{A} = \boldsymbol{A}_{(\ell+1)}$ fixed, updating $\boldsymbol{X}$ consists of decreasing the objective

$$\tfrac{1}{4}\big\|\boldsymbol{Y} - |\mathcal{F}(\boldsymbol{X})|^2\big\|_F^2 + \tfrac{\mu}{2}\big\|\mathcal{E}(\boldsymbol{X}) - \boldsymbol{D}\boldsymbol{A}\big\|_F^2 \tag{5}$$
$$\text{with} \quad \boldsymbol{X} \in \mathcal{X} = [0,1]^{N_1 \times N_2}.$$

This problem can be seen as a regularized version of the phase retrieval problem (with regularization parameter μ) that encourages the patches of $\boldsymbol{X}$ to be close to the sparse approximation $\boldsymbol{D}\boldsymbol{A}$ obtained during the previous iterate.

Our approach to decrease the value of the objective (5) is by a simple projected gradient descent step. In fact, for $\mu = 0$, this reduces to the so-called *Wirtinger Flow* method [11], but with necessary modifications to take into account the constraints on $\boldsymbol{X}$ (real-valuedness and variable bounds $[0,1]$).

The gradient (matrix) of $\varphi(\boldsymbol{X}) \coloneqq \frac{1}{4}\|\boldsymbol{Y} - |\mathcal{F}(\boldsymbol{X})|^2\|_F^2$ with respect to $\boldsymbol{X}$ can be computed as

$$\nabla\varphi(\boldsymbol{X}) = \Re\Big(\mathcal{F}^*\big(\mathcal{F}(\boldsymbol{X}) \odot (|\mathcal{F}(\boldsymbol{X})|^2 - \boldsymbol{Y})\big)\Big),$$

by using the chain rule (we omit a more detailed derivation due to space limitations). The gradient of $\psi(\boldsymbol{X}) \coloneqq \frac{\mu}{2}\|\mathcal{E}(\boldsymbol{X}) - \boldsymbol{D}\boldsymbol{A}\|_F^2$ is given by

$$\nabla\psi(\boldsymbol{X}) = \mu\mathcal{E}^*\big(\mathcal{E}(\boldsymbol{X}) - \boldsymbol{D}\boldsymbol{A}\big) = \mu\boldsymbol{R} \odot \mathcal{R}\big(\mathcal{E}(\boldsymbol{X}) - \boldsymbol{D}\boldsymbol{A}\big),$$

where $\boldsymbol{R} \in \mathcal{X}$ has entries r_{ij} equal to the number of patches the respective pixel x_{ij} is contained in. Note that if the whole image is divided into a complete set of non-overlapping patches, $\boldsymbol{R}$ will just be the all-ones matrix; otherwise, the element-wise multiplication with $\boldsymbol{R}$ undoes the averaging of pixel values performed by $\mathcal{R}$ when assembling an image from overlapping patches.

Finally, the gradient w.r.t. $\boldsymbol{X}$ of the objective in (5) is $\nabla\varphi(\boldsymbol{X}) + \nabla\psi(\boldsymbol{X}) \in \mathbb{R}^{N_1 \times N_2}$, and the update in Step 3 of Algorithm 1 is indeed shown to be a projected gradient descent step. Typically, a backtracking strategy may be used for choosing the step size γ_ℓ^X, as detailed in the section devoted to numerical experiments.

3.3. Updating the Dictionary

To update the dictionary, i.e., to approximately solve (3) w.r.t. $\boldsymbol{D}$ alone, keeping $\boldsymbol{X}$ and $\boldsymbol{A}$ fixed at their current values, we employ one pass of a block-coordinate descent (BCD) algorithm on the columns of the dictionary [21]. The objective to decrease may be written as

$$\sum_{i=1}^{p} \left\| \boldsymbol{D}\boldsymbol{a}^i_{(\ell+1)} - \boldsymbol{x}^i_{(\ell+1)} \right\|_2^2 \quad \text{s.t.} \quad \boldsymbol{D} \in \mathcal{D}, \tag{6}$$

and the update rule given by Steps 4 –13 corresponds exactly to one iteration of [19, Algorithm 11] applied to (6), which ensures the monotonic decrease of the objective function.

4. NUMERICAL EXPERIMENTS

We first describe some aspects of the concrete models and our implementation and then proceed to discuss our experiments.

4.1. Experimental Setup

We consider several linear operators $\mathcal{F}$ corresponding to different types of measurements. We denote by $\boldsymbol{F}$ the (normalized) 2D-Fourier transform, and introduce two complex Gaussian matrices $\boldsymbol{G} \in \mathbb{C}^{M_1 \times N_1}$, $\boldsymbol{H} \in \mathbb{C}^{M_2 \times N_2}$, whose entries are i.i.d. samples from the distribution $\mathcal{N}(0, \boldsymbol{I}/2) + i\mathcal{N}(0, \boldsymbol{I}/2)$. Then, we experiment with the operators $\mathcal{F}(\boldsymbol{X}) = \boldsymbol{G}\boldsymbol{X}$, $\mathcal{F}(\boldsymbol{X}) = \boldsymbol{G}\boldsymbol{X}\boldsymbol{G}^*$, $\mathcal{F}(\boldsymbol{X}) = \boldsymbol{G}\boldsymbol{X}\boldsymbol{H}^*$, and the coded diffraction model

$$\mathcal{F}(\boldsymbol{X}) = \begin{pmatrix} \boldsymbol{F}\big(\overline{\boldsymbol{M}}_1 \odot \boldsymbol{X}\big) \\ \vdots \\ \boldsymbol{F}\big(\overline{\boldsymbol{M}}_m \odot \boldsymbol{X}\big) \end{pmatrix}, \quad \mathcal{F}^*(\boldsymbol{Z}) = \sum_{j=1}^{m} \big(\boldsymbol{M}_j \odot \boldsymbol{F}^*(\boldsymbol{Z})\big), \tag{7}$$

where the $\boldsymbol{M}_j$'s are admissible coded diffraction patterns (CDPs), see for instance [11, Sect. 4.1]; in our experiments we used ternary CDPs, such that each $\boldsymbol{M}_j$ is in $\{0, \pm 1\}^{N_1 \times N_2}$. Note that the operator $\boldsymbol{F}$ is implemented using fast Fourier transforms.

To reconstruct $\hat{\boldsymbol{X}}$, we choose an oversampling setting where $M_1 = 4N_1$, $M_2 = 4N_2$ and/or $m = 2$, respectively. Moreover, we corrupt our measurements with additive white Gaussian noise $\boldsymbol{N}$ such that $\mathrm{SNR}(\boldsymbol{Y}, |\mathcal{F}(\hat{\boldsymbol{X}})|^2 + \boldsymbol{N}) = 10\,\mathrm{dB}$ for the Gaussian-type, and 20 dB for CDP measurements, respectively. Note that this settings yields a quite heavy noise level for the Gaussian cases, and a relatively low oversampling ratio for the CDPs.

4.2. Implementation Details

We choose to initialize our algorithm with a simple random image $\boldsymbol{X}_{(0)}$ in $\mathcal{X}$ to demonstrate the robustness of our approach with respect to its initialization. Nevertheless, other choices are possible. For instance, one may also initialize $\boldsymbol{X}_{(0)}$ with a power-method scheme similar to that proposed in [11], modified to account for the required real-valuedness and box-constraints. The dictionary is initialized as $\boldsymbol{D}_{(0)} = (\boldsymbol{I}, \boldsymbol{F}_D)$ in $\mathbb{R}^{s \times 2s}$, where $\boldsymbol{F}_D$ is is the 2D discrete cosine transform (see, e.g., [18]).

To update $\boldsymbol{A}$, we use the ISTA implementation from the SPAMS package [21] with its integrated backtracking line search (for L_A).[1] Regarding the step sizes γ_ℓ^X for the update of $\boldsymbol{X}$ (Step 3 of Alg. 1), we adopt the following simple strategy: Whenever the gradient step leads to a reduction in the objective function value, we accept it; otherwise, we recompute the step with γ_ℓ^X halved until a reduction was achieved (no more than 100 trials). Regardless of whether γ_ℓ^X was reduced or not, we reset its value to $1.68\gamma_\ell^X$ for the next round; the initial step size is 1.

Finally, we consider non-overlapping 8×8 patches (though our code can also handle overlap). We run PRDL (Algorithm 1) with $K_1 = 25$ and $K_2 = 50$; the regularization/penalty parameter values can be read from Table 1 (there, m_Y is the number of elements of $\boldsymbol{Y}$). We remark that these parameter values were empirically found to work well for the instances considered here but are not the outcome of rigorous parameter benchmarking, which is a subject of future investigations.

4.3. Computational Experiments

We test our method on a collection of typical (grayscale) test images used in the literature, namely cameraman, house and peppers of size 256×256, and lena, barbara, boat, fingerprint and mandrill of size 512×512. All experiments were carried out on a Linux 64-bit quad-core machine (2.8 GHz, 8 GB RAM) running Matlab R2015a.

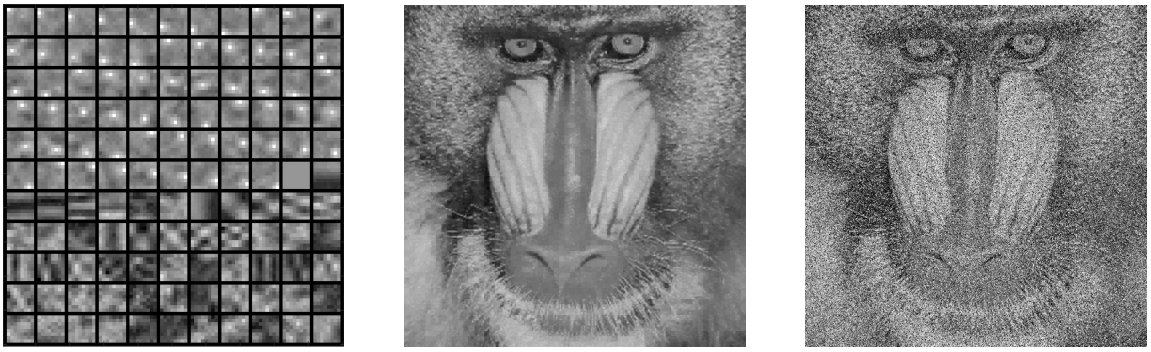
We evaluate our approach with the following question in mind: *Can we improve upon the quality of reconstruction compared to standard phase retrieval algorithms?* Standard methods cannot exploit sparsity if the underlying basis or dictionary is unknown; as we will see, the introduced (patch-)sparsity indeed allows for better recovery results (at least in the oversampling and noise regimes considered here).

To evaluate the achievable sparsity, we look at the average number of nonzeros in the columns of $\boldsymbol{A}$ after running our algorithm. Generally, smaller values indicate an improved suitability of the learned dictionary for sparse patch coding (high values often occur if the regularization parameter λ is too small and the dictionary is learning the noise, which is something we would like to avoid). To assess the quality of the image reconstruction, we consider two standard measures, namely the peak signal-to-noise ratio (PSNR) of the reconstructed $\boldsymbol{X}$ as well as its structural similarity index (SSIM) [26]. For PSNR, larger values are better, and SSIM-values closer to 1 (always ranging between 0 and 1) indicate better visual quality.

Table 1 displays the CPU times, PSNR- and SSIM-values and mean patch representation vector sparsity levels obtained for the various measurement types, averaged over the instance groups of the same size. The concrete example in Figure 1 shows the results from PRDL and plain Wirtinger Flow (WF; the real-valued, $[0, 1]$-box constrained variant, which corresponds to running Algorithm 1 with $\mu = 0$ and omitting the updates of $\boldsymbol{A}$ and $\boldsymbol{D}$) for CDP measurements of the mandrill image. (In all tests, we let the Wirtinger Flow method run for the same number of iterations (75) and use the same starting points as for the PRDL runs.)

The PRDL method consistently provides better image reconstructions than WF, which clearly shows that our approach successfully introduces sparsity into the phase retrieval problem and exploits

[1] `http://spams-devel.gforge.inria.fr/`.



(a) (b) (c)

Fig. 1. PRDL example: Image original is the 512×512 "mandrill" picture, measurements are noisy CDPs (obtained using two ternary masks). (a) final dictionary (excerpt), (b) image reconstruction $\mathcal{R}(\boldsymbol{DA})$ from sparsely coded patches, (c) reconstruction $\boldsymbol{X}_{\text{WF}}$ after 75 WF iterations. Final PSNR 25.39 dB ($\boldsymbol{X}_{\text{PRDL}}$: 26.45, $\boldsymbol{X}_{\text{WF}}$: 13.42) and SSIM 0.7212 ($\boldsymbol{X}_{\text{PRDL}}$: 0.7601, $\boldsymbol{X}_{\text{WF}}$: 0.1562), average $\|\boldsymbol{a}^i\|_0$ is 11.95.

		256×256 instances					512×512 instances				
$\mathcal{F}$ type	reconstruction	$(\mu, \lambda)/m_Y$	time	PSNR	SSIM	avg. $\|\boldsymbol{a}^i\|_0$	$(\mu, \lambda)/m_Y$	time	PSNR	SSIM	avg. $\|\boldsymbol{a}^i\|_0$
$\boldsymbol{G}\hat{\boldsymbol{X}}$	$\boldsymbol{X}_{\text{PRDL}}$	(0.5,0.068)	10.85	24.75	0.5370		(0.5,0.068)	52.93	21.18	0.5592	
	$\mathcal{R}(\boldsymbol{DA})$			24.66	0.6330	7.02			18.17	0.4964	2.12
	$\boldsymbol{X}_{\text{WF}}$		5.74	18.74	0.2812	–		29.72	18.80	0.3773	–
$\boldsymbol{G}\hat{\boldsymbol{X}}\boldsymbol{G}^*$	$\boldsymbol{X}_{\text{PRDL}}$	(0.5,0.226)	42.55	22.42	0.4042		(0.5,0.068)	238.91	22.54	0.5275	
	$\mathcal{R}(\boldsymbol{DA})$			23.22	0.7270	6.72			23.82	0.7679	12.01
	$\boldsymbol{X}_{\text{WF}}$		32.53	22.41	0.4041	–		199.13	22.54	0.5274	–
$\boldsymbol{G}\hat{\boldsymbol{X}}\boldsymbol{H}^*$	$\boldsymbol{X}_{\text{PRDL}}$	(0.5,0.226)	41.92	22.62	0.4114		(0.5,0.068)	242.71	22.53	0.5269	
	$\mathcal{R}(\boldsymbol{DA})$			23.24	0.7337	6.51			23.83	0.7688	11.99
	$\boldsymbol{X}_{\text{WF}}$		33.23	22.61	0.4111	–		202.40	22.53	0.5268	–
CDP (cf. (7))	$\boldsymbol{X}_{\text{PRDL}}$	(0.05,0.003)	7.51	27.20	0.7434		(0.05,0.003)	31.17	27.16	0.7764	
	$\mathcal{R}(\boldsymbol{DA})$			26.62	0.7675	7.79			26.18	0.7605	11.85
	$\boldsymbol{X}_{\text{WF}}$		2.29	13.03	0.1147	–		10.64	13.01	0.1539	–

Table 1. Test results for m_Y Gaussian-type and coded diffraction pattern (CDP) measurements. Reported are mean values (geometric mean for CPU times) per measurement type over the three 256×256 and five 512×512 instances w.r.t. the reconstructions from PRDL ($\boldsymbol{X}_{\text{PRDL}}$ and $\mathcal{R}(\boldsymbol{DA})$) with parameters (μ, λ) and (real-valued, $[0, 1]$-constrained) Wirtinger Flow ($\boldsymbol{X}_{\text{WF}}$), resp. (CPU times in seconds, PSNR in dB).

it in the solution process. As can be seen from Table 1, the obtained dictionaries allow for rather sparse representation vectors, with the effect of making better use of the information provided by the measurements, and also denoising the image along the way. The latter fact can be seen in the example (Fig. 1) and also inferred from the significantly higher PSNR and SSIM values for the estimates $\boldsymbol{X}_{\text{PRDL}}$ and $\mathcal{R}(\boldsymbol{DA})$ obtained from PRDL compared to the reconstruction $\boldsymbol{X}_{\text{WF}}$ of the WF algorithm (which does not make use of sparsity).

5. DISCUSSION AND CONCLUSION

We believe that our experiments demonstrate that dictionary learning for phase retrieval with a patch-based sparsity is a promising direction, especially for cases where the original Wirtinger Flow approach fails. For the future, we are planning to conduct larger-scale experiments to answer some questions that we have left open.

For example, we expect further improvements by rigorously benchmarking the algorithmic parameters (in particular, to find good default settings). Several variants of our algorithm may also be developed—for instance, we successfully used ℓ_0-constraints instead of the ℓ_1-penalty, by combining Orthogonal Matching Pursuit (OMP) [27] with our framework. Finally, to evaluate the quality of the learned dictionary, one might also ask how PRDL compares with the straightforward approach to first run (standard) phase retrieval and then learn dictionary and sparse patch representations from the result. Some preliminary experiments (not reported here) indicate that both approaches produce comparable results in the noisefree setting, while our numerical results demonstrate a denoising feature of our algorithm that the simple approach would obviously lack. Another interesting aspect to evaluate is by how much reconstruction quality and achievable sparsity degrade due to the loss of phase (or, more generally, measurement nonlinearity).

Acknowledgements

The work of J. Mairal was funded by the French National Research Agency [Macaron project, ANR-14-CE23-0003-01]. The work of Y. Eldar was funded by the European Union's Horizon 2020 research and innovation programme under grant agreement ERC-BNYQ, and by the Israel Science Foundation under Grant no. 335/14.

6. REFERENCES

[1] S. Marchesini, "A Unified Evaluation of Iterative Projection Algorithms for Phase Retrieval," *Review of Scientific Instruments*, vol. 78, pp. 011301–1–10, 2007.

[2] Y. Shechtman, Y. C. Eldar, O. Cohen, H. N. Chapman, J. Miao, and M. Segev, "Phase Retrieval with Application to Optical Imaging," *IEEE Signal Process. Mag.*, vol. 32, no. 3, pp. 87–109, 2015.

[3] R. W. Gerchberg and W. O. Saxton, "A practical algorithm for the determination of the phase from image and diffraction plane pictures," *Optik*, vol. 35, no. 2, pp. 237–246, 1972.

[4] J. R. Fienup, "Phase Retrieval Algorithms: A Comparison," *Appl. Opt.*, vol. 21, no. 15, pp. 2758–2769, 1982.

[5] H. H. Bauschke, P. L. Combettes, and D. R. Luke, "Phase retrieval, error reduction algorithm, and Fienup variants: A view from convex optimization," *J. Opt. Soc. Amer. A*, vol. 19, no. 7, pp. 1334–1345, 2002.

[6] E. J. Candés, Y. C. Eldar, T. Strohmer, and V. Voroninski, "Phase retrieval via matrix completion," *SIAM J. on Imaging Sciences*, vol. 6, no. 1, pp. 199–225, 2013.

[7] Y. Shechtman, Y. C. Eldar, A. Szameit, and M. Segev, "Sparsity based sub-wavelength imaging with partially incoherent light via quadratic compressed sensing," *Optics Express*, vol. 19, no. 16, pp. 14807–14822, 2011.

[8] K. Jaganathan, S. Oymak, and B. Hassibi, "Recovery of sparse 1-D signals from the magnitudes of their Fourier transform," arXiv:1206.1405[cs.IT], 2012.

[9] H. Ohlsson, A. Y. Yang, R. Dong, and S. S. Sastry, "Compressive Phase Retrieval From Squared Output Measurements Via Semidefinite Programming," arXiv:1111.6323[math.ST], 2012.

[10] I. Waldspurger, A. d'Aspremont, and S. Mallat, "Phase recovery, MaxCut and complex semidefinite programming," *Math. Prog. A*, vol. 149, no. 1, pp. 47–81, 2015.

[11] E. J. Candés, X. Li, and M. Soltanolkotabi, "Phase Retrieval via Wirtinger Flow: Theory and Algorithms," arXiv:1407.1065[cs.IT], 2014, to appear in IEEE Trans. Inf. Theory.

[12] T. T. Cai, X. Li, and Z. Ma, "Optimal Rates of Convergence for Noisy Sparse Phase Retrieval via Thresholded Wirtinger Flow," arXiv:1506.03382[math.ST], 2015.

[13] Y. Shechtman, A. Beck, and Y. C. Eldar, "GESPAR: Efficient Phase Retrieval of Sparse Signals," *IEEE Trans. Signal Process.*, vol. 62, no. 4, pp. 928–938, 2014.

[14] M. L. Moravec, J. K. Romberg, and R. G. Baraniuk, "Compressive Phase Retrieval," in *Proc. SPIE 6701: Wavelets XII*, D. Van De Ville, V. K. Goyal, and M. Papadakis, Eds. 2007, pp. 670120–1–11, International Society for Optical Engineering.

[15] S. Foucart and H. Rauhut, *A Mathematical Introduction to Compressive Sensing*, Applied and Numerical Harmonic Analysis. Birkhäuser, 2013.

[16] Y. C. Eldar and G. Kutyniok, Eds., *Compressed Sensing: Theory and Applications*, Cambridge University Press, 2012.

[17] B. A. Olshausen and D. J. Field, "Emergence of Simple-Cell Receptive Field Properties by Learning a Sparse Code for Natural Images," *Nature*, vol. 381, pp. 607–609, 1996.

[18] M. Elad and M. Aharon, "Image Denoising Via Sparse and Redundant Representations Over Learned Dictionaries," *IEEE Trans. Image Process.*, vol. 15, no. 12, pp. 3736–3745, 2006.

[19] J. Mairal, F. Bach, and J. Ponce, "Sparse Modeling for Image and Vision Processing," *Foundations and Trends in Computer Graphics and Vision*, vol. 8, no. 2-3, pp. 85–283, 2014.

[20] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4311–4322, 2006.

[21] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *Journal of Machine Learning Research*, vol. 11, pp. 19–60, 2010.

[22] A. Beck and M. Teboulle, "A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems," *SIAM J. Imaging Sci.*, vol. 2, no. 1, pp. 183–202, 2009.

[23] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic Decomposition by Basis Pursuit," *SIAM J. Sci. Comput.*, vol. 20, no. 1, pp. 33–61, 1998.

[24] A. M. Tillmann, "On the Computational Intractability of Exact and Approximate Dictionary Learning," *IEEE Signal Process. Lett.*, vol. 22, no. 1, pp. 45–49, 2015.

[25] P. Tseng and S. Yun, "A coordinate gradient descent method for nonsmooth separable minimization," *Math. Prog. B*, vol. 117, no 1, pp. 387–423, 2009.

[26] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image Quality Assessment: From Error Visibility to Structural Similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, 2004.

[27] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal Matching Pursuit: Recursive Function Approximation with Applications to Wavelet Decomposition," in *Proc. 27th Annual Asilomar Conference on Signals, Systems and Computers*. 1993, vol. 1, pp. 40–44, IEEE Computer Society Press.